2001-12

A Lagrangian Heuristic for solving a network interdiction problem

Bingol, Levent

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A LAGRANGIAN HEURISTIC FOR SOLVING A NETWORK INTERDICTION PROBLEM

by

Levent Bingol

December 2001

Thesis Advisor:	R. Kevin Wood
Second Reader:	Kelly J. Cormican

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Title (Mix case letters) A Lagrangian Heuristic for Solving a Network Interdiction Problem			**5. FUNDING NUMBERS**
6. AUTHOR(S) Bingol, Levent			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

This thesis is concerned with solving or approximately solving a maximum-flow network-interdiction problem denoted MXFI: A network user strives to maximize flow of a commodity through a capacitated network, while an interdictor, with limited assets, attempts to destroy links in the network to minimize that maximum flow.

MXFI can be converted to a binary integer program and solved but this approach can be computationally expensive. Earlier work by Derbes (1997) on a Lagrangian-relaxation technique has shown promise for solving the problem more quickly (Derbes, 1997). We extend his technique and implement algorithms in C to solve MXFI for all integer values of total interdiction resource available, *R*, in some specified range; interdictable arcs require one unit of resource to destroy. The basic procedure solves MXFI exactly for most values of *R*, but "problematic values" of *R* do arise. For one set of test problems, a heuristic handles these values successfully, with optimality gaps that are typically less than three percent.

We test our algorithms and implementations using five test networks which range in size from 27 nodes and 86 arcs to 402 nodes and 1826 arcs. Using a 700 MHz Pentium III personal computer, we solve the largest problem in 16 seconds.

14. SUBJECT TERMS Network Interdiction, Mathematical Modeling, Lagrangian Relaxation			**15. NUMBER OF PAGES** 54
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

A LAGRANGIAN HEURISTIC FOR SOLVING A NETWORK INTERDICTION PROBLEM

Levent Bingol
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1996

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2001

Author: _____
Levent Bingol

Approved by: _____
R. Kevin Wood, Thesis Advisor

_____
Kelly J. Cormican , Second Reader

_____
James N. Eagle, Chairman
Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis is concerned with solving or approximately solving a maximum-flow network-interdiction problem denoted MXFI: A network user strives to maximize flow of a commodity through a capacitated network, while an interdictor, with limited assets, attempts to destroy links in the network to minimize that maximum flow.

MXFI can be converted to a binary integer program and solved but this approach can be computationally expensive. Earlier work by Derbes (1997) on a Lagrangian-relaxation technique has shown promise for solving the problem more quickly (Derbes, 1997). We extend his technique and implement algorithms in C to solve MXFI for all integer values of total interdiction resource available, *R*, in some specified range; interdictable arcs require one unit of resource to destroy. The basic procedure solves MXFI exactly for most values of *R*, but "problematic values" of *R* do arise. For one set of test problems, a heuristic handles these values successfully, with optimality gaps that are typically less than three percent.

We test our algorithms and implementations using five test networks which range in size from 27 nodes and 86 arcs to 402 nodes and 1826 arcs. Using a 700 MHz Pentium III personal computer, we solve the largest problem in 16 seconds.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I thank my advisor Professor Kevin Wood for his invaluable advice, guidance and patience during the course of this research. I have enjoyed exploring this area of Operations Research with him.

I thank LCDR Kelly Cormican for his flexibility and support as a second reader. His encouragement provided me great motivation throughout this research.

I thank to Turkish Navy for giving me the opportunity to study at the Naval Post Graduate School.

Finally, I thank to my mother, Nihal Bingol, and my father, Iznullah Bingol, whose love has been a constant inspiration and a great motivation to me throughout my life. Thanks for growing me as a free spirit.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Consider a theater of war in which one warring party uses a transportation network to provide supplies, troops, and ammunition to his forces and suppose that the opposing party has a limited ability to attack and disrupt his opponent's use of that network. This thesis is concerned with solving or approximately solving a related maximum-flow network-interdiction problem denoted MXFI: An interdictor, with limited assets, attempts to destroy links in a network in order to minimize the maximum flow through that network which an adversary might obtain. The problem is described by Wood (1993) and was extended for use at the U.S. Strategic Command by Whiteman (1999).

This thesis is motivated by the possibility of weakening a military force by disrupting its access to supplies, but other uses may exist. For example, we may wish to disrupt the escape routes of a fugitive or reduce the flow of illegal drugs and precursor chemicals moving through a network of rivers, road and air corridors. Indeed, MXFI was originally motivated by the United States' drug interdiction efforts in South America.

MXFI can be converted to a binary integer program and solved but this approach can be computationally expensive. Furthermore, the approach does not lend itself to efficient exploration of tradeoffs between interdiction resource expenditure and maximum post-interdiction flow. However, earlier work by Derbes (1997) on a Lagrangian-relaxation technique has shown promise for solving MXFI more quickly and facilitating the exploration of such tradeoffs. We extend Derbes' techniques in this thesis.

In the Lagrangian-relaxation technique, for fixed resource level, we relax the interdiction resource constraint in the basic integer program using a parameter λ. This relaxation allows us to approximately solve the problem by moving the resource constraint into the objective function. The resulting problem is almost as easy to solve as if resource constraints were ignored: It is merely a maximum-flow problem. We extend this technique and implement algorithms in C to solve MXFI for all integer values of

total interdiction resource available, R, in some specified range. Interdictable arcs are assumed to require one unit of resource to destroy. The basic procedure solves MXFI exactly for most values of R, but "problematic values" of R do arise. A heuristic usually handles these values successfully: For one set of test problems, the heuristic yields relative optimality gaps that are typically less than three percent. Relative gaps are typically small, a few percent, but can be large if the post-interdiction maximum flow is small.

We test our algorithm, coded in C, using five test networks ranging in size from 27 nodes and 86 arcs to 402 nodes and 1826 arcs. All tests are performed on a 700 MHz Pentium III personal computer with The Microsoft Windows Millennium operating system and Microsoft Visual C++ compiler. The largest network is solved in 16 seconds for 58 potential values of R.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In the maximum-flow network-interdiction problem (MXFI), an "interdictor" attempts to destroy parts of a capacitated network using limited interdiction resources, so as to minimize the maximum flow that a "network user" can move through the network. This thesis investigates an efficient Lagrangian-relaxation technique that solves MXFI, at least approximately, for all values of a single interdiction resource in a specified range.

A. OVERVIEW

This thesis is concerned with solving or approximately solving a maximum-flow network-interdiction problem described by Steinrauf (1991) and Wood (1993), and extended for use at the U.S. Strategic Command by Whiteman (1999). The problem is defined on a directed network $G = (N,A)$ which has arc capacities u_k for all arcs $k \in A$, a source node s, a sink node t, and an artificial "return arc" $a = (t,s)$ included in A such that $u_a = \infty$. We are interested in solving he problem for various amounts of interdiction resource R, so we parameterize the problem by R:

$$\text{MXFI}(R) \qquad z(R) = \min_{\mathbf{x}\in X} \max_{\mathbf{y}} y_a$$

$$\text{s.t.} \sum_{k\in FS(i)} y_k - \sum_{k\in RS(i)} y_k = 0 \qquad \forall i \in N$$

$$0 \le y_k \le u_k(1-x_k) \qquad \forall k \in A$$

where $X = \left\{ \mathbf{x} \in \{0,1\}^{|A|} \,\middle|\, \sum_{k\in A} r_k x_k \le R,\ x_a \equiv 0 \right\}$, r_k is the amount of resource required to interdict arc k, R is the total amount of resource available, $FS(i)$ (forward star of node i) is the set of arcs directed out of node i, and $RS(i)$ (reverse star of node i) is the set of arcs directed into node i. For fixed $\mathbf{x}$, the inner maximization problem is a standard maximum-flow problem with arc capacities $u_k(1-x_k)$. But, the interdictor controls the $\mathbf{x}$ variables and attempts to minimize that maximum flow: When $x_k = 1$, arc k is interdicted and its capacity goes to 0; otherwise, the arc is left untouched and takes on its normal

capacity u_k. We consider only the case in which all r_k are integer. Thus, R may be assumed to be integer.

MXFI(R) can be converted to a binary integer program and solved using standard techniques, but this approach can be computationally expensive (Wood, 1993). Earlier work by Derbes (1997) on a Lagrangian-relaxation technique has shown promise for efficient solutions, but he only solves MXFI(R) for fixed R and we believe planners will often want to explore the effects of varying R. That is, they will want to understand how increasing interdiction effort results in improved interdiction results over a range of interdiction resource values. Therefore, we adapt Derbes' method to solve MXFI(R) over a wide range of values R. In particular, we solve over the widest range of "sensible" values, which are from $R = 0$ to the smallest value of R that results in a maximum flow of zero through the network. Our method typically solves MXFI(R) exactly for most values of R, but not for a few "problematic" ones. We provide an effective heuristic to deal with these problematic instances.

B. LITERATURE SEARCH

The study of network interdiction began during the Vietnam War with efforts to destroy enemy supply lines (Wollmer 1964, 1970). Later, the desire to interdict illicit drugs generated additional interest; see Steinrauf (1991) and (Phillips 1992).

MXFI can be viewed as having evolved over time through the studies of Wollmer (1964, 1970), Durbin (1966), McMasters and Mustin (1970), Helmbold (1971), Ghare, Montgomery, and Turner (1971), Lubore, Ratliff and Sicilia (1971, 1975), Steinrauf (1991), Phillips (1992) and Wood (1993). Most of these papers are based on the pioneering work on maximum flows by Ford and Fulkerson (1956).

Steinrauf (1991) solves the network-interdiction problem with mathematical-programming techniques. Two mathematical programs are developed which determine strategies to interdict a network using limited resources. The first model identifies a set of arcs whose interdiction minimizes the maximum flow through the network while not exceeding available interdiction resources. This is essentially MXFI. The second model identifies a set of arcs whose interdiction isolates a large set of nodes around a specified node, which might represent the most likely location of a drug laboratory. The models

are applied to a sample network that is similar to a river and road network in Bolivia where counter-narcotic interdiction operations were being conducted at the time.

Philips (1992) presents several pseudo-polynomial time algorithms using dynamic programming for interdiction of undirected planar networks.

Wood (1993) develops integer-programming models for MXFI and variants and shows MXFI to be NP-complete.

Reed (1994) derives an integer-programming model to maximize the longest path in a PERT network through interdiction. The purpose is to slow the advance of nuclear-weapons technology in adversarial countries.

Wollmer (1970) and Washburn and Wood (1994) study game-theoretic network interdiction models, which are inappropriate for the problem we consider.

Cormican (1995) solves MXFI using Benders decomposition and improves computational speed with a "flow-dispersion heuristic." Cormican begins the investigation of probabilistic versions of MXFI where interdiction successes or arc capacities are uncertain. Cormican, Morton and Wood (1996) develop this topic further and solve such problems with a sequential-approximation algorithm.

Derbes (1997) follows a mathematical-programming approach for solving MXFI. He shows that a technique based on Lagrangian relaxation can be effective in approximately solving the problem. It turns out that the network interdictor's problem of minimizing the maximum flow through the network is difficult to solve because of the interdiction budget constraint. Therefore, he relaxes this constraint using Lagrangian relaxation, which allows the interdictor to violate the constraint while paying a penalty. For a fixed value of a penalty parameter, the relaxation is an easy-to-solve maximum-flow problem with a solution that provides a lower bound on the optimal solution to the original problem. A corresponding interdiction solution is also derived although this may or may not be feasible. He maximizes the lower bound using binary search on the value of the penalty parameter, solving a maximum flow problem at each step and guaranteeing that at least one feasible solution will be found. The best feasible solution obtained in this process is taken as an approximate, possibly optimal, solution to the problem, and the corresponding upper bound is compared to the maximized lower bound to judge solution

quality. Derbes implements the procedure in C and demonstrates its effectiveness. (He also studies, with less success, a dynamic version of MXFI where flow requires time to pass through a network's arcs.)

Golden (1978), Israeli (1999), Israeli and Wood (1999) and Wevley (1999) study another category of network interdiction problem. In this problem, an interdictor attempts to interdict (destroy or lengthen) arcs, using limited interdiction assets, to maximize the length of a shortest *s*-*t* path. While these models are similar in spirit to MXFI, that research is not relevant to this thesis.

Whiteman (1999) presents a comprehensive strategy to plan the interdiction of complex infrastructure networks that can be modeled as capacitated flow networks. The strategy is designed to achieve a high level of interdiction in a single set of strikes, reducing the necessity of costly follow-up strikes. He adapts the integer program of Wood (1993) to select target sets of minimum size and uses ad hoc Monte Carlo techniques to handle uncertain interdiction effects.

Akgun (2000) studies the "*K*-group network interdiction problem" in which a network user attempts to maximize flow among three or more groups of nodes, while an interdictor interdicts network arcs, using limited interdiction resources, to minimize that flow. This model is more general than MXFI but an extension of our techniques might be of value in solving that problem.

C. OUTLINE OF THESIS

The first chapter of the thesis has introduced the max-flow network-interdiction model (MXFI), and given an overview of earlier research in this area. Chapter II provides definitions, notation and detailed background on MXFI, and describes some other basic models. In Chapter III, we develop our Lagrangian-relaxation approach, which attempts to solve MXFI for every value of "reasonable" *R*, and describe a heuristic that is added to deal with problematic values of *R*. Chapter IV provides computational results. Conclusions and recommendations for future work are covered in Chapter V.

II. PRELIMINARIES

This chapter provides essential definitions and notation for MXFI(R), and outlines the derivation of Derbes' Lagrangian-relaxation technique for solving MXFI (Derbes 1997). The notation, conventions and models follow Wood (1993), Cormican (1995) and Derbes (1997).

A. DEFINITIONS AND NOTATION

MXFI(R) is defined on a directed network $G = (N,A)$ where N is a set of n nodes and A is a set of m directed arcs. An arc $k = (i,j)$ originates from tail node i and terminates at head node j. The forward star of node i, denoted $FS(i)$, is the set of arcs directed out of node i and the reverse star of node i, denoted $RS(i)$, is the set of arcs directed into node i.

The network $G = (N,A)$ has arc capacities $u_k > 0$, a source node s, a sink node t, and an artificial return arc $a = (t,s)$ included in A such that $u_a = \infty$. An interdictor has a total of R units of resource available for interdiction.

A single type of interdiction is assumed in this thesis and only arcs are interdicted. Any "interdictable arc" k requires $r_k > 0$ units of resource to interdict; r_k is assumed to be integral and thus R may also be assumed to be integral. "Uninterdictable arcs," which always include $k = a$, have $r_k = \infty$. These arcs may not be interdicted at any cost for political, tactical, theoretical or other reasons. Standard network transformations enable the modeling of undirected networks and node interdiction if desired (e.g., Ahuja et al. 1997, pp. 38, 46). Other generalizations such as "partial interdiction'' (Wood 1993) are possible, too.

A cut (N_s, N_t), also denoted C, is a partition of the node set N into two subsets, N_s and N_t, such that $s \in N_s$ and $t \in N_t$. Each cut defines a set of arcs that have one endpoint in N_s and other endpoint in N_t. With respect to the cut, an arc $k=(i,j)$ is a forward arc if $i \in N_s$ and $j \in N_t$; otherwise it is backward arc. The set of forward arcs is denoted by A_C, and we often refer to this set as "the cut'' for the sake of simplicity. The capacity of the cut is $\sum_{k \in A_C} u_k$. A minimum cut is a cut whose capacity is minimum among all possible

cuts in the network. The maximum-flow minimum-cut theorem (Ford and Fulkerson 1956) states that the maximum flow *s*-*t* flow in a network *G* equals the capacity of a minimum cut.

B. MAXIMUM-FLOW NETWORK INTERDICTION MODELS

1. Maximum-Flow Network-Interdiction Model Reformulation

Cormican (1995) reformulates MXFI(*R*) for easy conversion to a MIP:

MXFI-RF(*R*):

Indices:

$i, j \in N$ nodes of directed network $G = (N,A)$ including two special nodes, the source *s* and the sink *t*

$k \in A$ directed arcs in the network $G = (N,A)$, $k = (i,j)$

Data:

u_k nominal capacity of arc *k*

r_k amount of resource required to interdict arc *k*

R total amount of resource available to the network interdictor

Decision Variables:

Network user:

y_k amount of flow on arc *k*

Network interdictor:

x_k 1 if arc *k* is interdicted; 0 otherwise

Formulation:

$$z(R) = \min_{\mathbf{x} \in X} \max_{\mathbf{y}} y_a - \sum_{k \in A} x_k y_k$$

s.t.

$$\sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k = 0 \qquad \forall i \in N$$

$$0 \le y_k \le u_k \qquad \forall k \in A$$

where $X = \left\{ \mathbf{x} \in \{0,1\}^{|A|} \,\middle|\, \sum_{k \in A} r_k x_k \le R,\ x_a \equiv 0 \right\}$.

2. Integer Program

Taking the dual of the inner maximization yields the following mixed-integer program first derived through other means by Wood (1993).

MXFI-IP(*R*):

Indices and Data: As in MXFI-RF(*R*)

Decision Variables:

x_k 1 if arc k is interdicted; 0 otherwise

α_i $\alpha_i = 1$ if $i \in N_t$, else $\alpha_i = 0$

β_k $\beta_k = 1$ if k is a forward arc of cut and not interdicted, else $\beta_k = 0$

Formulation:

$$z(R) = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{x}} \sum_{k \in A} u_k \beta_k$$

s.t.

$$\alpha_i - \alpha_j + x_k + \beta_k \ge 0 \qquad \forall k = (i,j) \in A - a$$

$$\alpha_t - \alpha_s + x_a + \beta_a \ge 1$$

$$\alpha_i \in \{0,1\} \qquad \forall i \in N$$

$$\alpha_s \equiv 0,\ \alpha_t \equiv 1$$

$$\beta_k \in \{0,1\} \qquad \forall k \in A$$

$$\beta_a \equiv 0$$

$$x_k \in \{0,1\} \qquad \forall k \in A$$

$$x_a \equiv 0$$

$$\sum_{k \in A} r_k x_k \le R \qquad (1)$$

The variables α_i identify a cut (N_s,N_t) with $i \in N_s$ if $\alpha_i = 0$ and $i \in N_t$ if $\alpha_i = 1$. The variables x_k and β_k represent interdiction decisions with respect to (N_s,N_t) and have the following interpretation: For forward arcs $k = (i,j)$ in the cut, $\alpha_i - \alpha_j = -1$ so $x_k + \beta_k = 1$ is required. So, either $x_k = 1$, indicating that this arc is interdicted, or $\beta_k = 1$, indicating that this arc is not interdicted and forms part of the minimum cut after interdiction. $x_k = \beta_k = 0$ indicates that arc k is neither interdicted nor part of the minimum cut after interdiction.

3. Lagrangian Relaxation for the Integer Problem

The basic integer program formulation MXFI-IP(R) is hard to solve but becomes easy if we ignore the interdiction budget constraint (1). While this constraint cannot be ignored, it can be relaxed. Following Derbes (1997), we use Lagrangian relaxation and moved the interdiction resource constraint into the objective using a parameter λ. The resulting problem is almost as easy to solve as if constraint (1) were ignored. The Lagrangian relaxation of MXFI-IP(R) is:

LR(λ,R):

$$z(\lambda, R) = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{x}} \sum_{k \in A} (u_k \beta_k + \lambda r_k x_k) - \lambda R$$

s.t. $\alpha_i - \alpha_j + x_k + \beta_k \ge 0 \qquad \forall k = (i,j) \in A - a$

$\alpha_t - \alpha_s + x_a + \beta_a \ge 1$

$\alpha_i \in \{0,1\} \qquad \forall i \in N$

$\alpha_s \equiv 0,\ \alpha_t \equiv 1$

$\beta_k \in \{0,1\} \qquad \forall k \in A$

$\beta_a \equiv 0$

$x_k \in \{0,1\} \qquad \forall k \in A$

$x_a \equiv 0$

It can be shown that the LP relaxation of this model has integer extreme points so we may take its dual and simplify to obtain:

LRD(λ,R):

$$z(\lambda, R) = \max_{\mathbf{y}} y_a - \lambda R$$

$$\text{s.t.} \sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k = 0 \qquad \forall i \in N$$

$$0 \le y_k \le \min\{u_k, \lambda r_k\} \qquad \forall k \in A$$

LRD(λ,R) is essentially a maximum flow model defined on G but using capacities modified by the dual cost of interdicting the arc.

Proposition 1: Any solution to LRD(λ,R) finds a minimum-cut (N_s, N_t) that corresponds to a feasible or infeasible solution ($\mathbf{x}$,α,β) to the original problem MXFI-IP(R) as follows:

1. $\alpha_i = 1 \ \forall i \in N_t$, $\alpha_i = 0 \ \ \forall i \in N_s$

2. $\beta_k = 1$ if $i \in N_s, j \in N_t$ and $y_k = u_k$, i.e., if k is a forward arc of the minimum cut and $u_k < \lambda r_k$, then k is not interdicted.

3. $x_k = 1$ if $i \in N_s, j \in N_t$ and $y_k = \lambda r_k$, i.e., if k is a forward arc of the minimum cut and $u_k > \lambda r_k$, then k is interdicted.

4. $x_k = \beta_k = 0 \ \forall k$ that are not forward arcs in the cut. ■

Note that λ can always be perturbed so that $u_k = \lambda r_k$ does not occur. The solution $\mathbf{x}$ is feasible if the interdiction budget constraint (1) holds.

THIS PAGE INTENTIONALLY LEFT BLANK

III. NETWORK INTERDICTION BY LAGRANGIAN RELAXATION

This chapter presents our approach to solving MXFI(R) through LRD(λ,R), for a range of resource values R.

A. OVERVIEW

We use LRD(λ,R) as our main model to develop the algorithms to solve MXFI(R). The function $z(\lambda,R)$, which LRD(λ,R) computes, is a piecewise-linear concave function in λ and Derbes (1997) shows, for a fixed value of R, how to maximize this function.

Because of the relaxation, $z(\lambda,R) \leq z(R)$, but for some λ values we may find a Lagrangian multiplier such that $\sum_{k\in A} r_k x_k = R$, where the x_k are computed via Proposition 1. In this case,

$$z(\lambda,R) = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{x}} \sum_{k\in A} u_k\beta_k + \lambda(\sum_{k\in A} r_k x_k - R) = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{x}} \sum_{k\in A} u_k\beta_k = z(R). \quad (2)$$

That is, we have solved MXFI(R) exactly. Derbes (1997) uses binary search on λ values to maximize $z(\lambda,R)$ for fixed R and often finds λ such that $z(\lambda,R) = z(R)$, but not always.

In this chapter, we adapt Derbes' method to attempt to solve MXFI(R) for all integer values of R over the largest meaningful range, which is from the smallest value that allows no positive post-interdiction flow down to R = 0. (In fact, we search this range from large to small.) For the sake of simplicity, we assume from hereon that r_k = 1 for all k; even this special case of MXFI is NP-complete (Wood 1993).

B. SOLVING THE RELAXED MODEL

Proposition 2: Given a fixed λ, with $\lambda \neq u_k$ for any k, we can solve MXFI(R), for some as yet unspecified R, through LRD(λ,R), as follows:

1. Find a minimum-capacity cut A_C in G with arc capacities $\min\{u_k, \lambda r_k\}\ \forall k \in A$ where $r_k = 1$ for all interdictable arcs k, and $r_k = \infty$, otherwise. This is easily accomplished with an efficient maximum-flow algorithm, which computes $f(\lambda)$ (e.g., Ahuja et al. 1993, Chap.7).

2. Arcs in the minimum-capacity cut A_C are examined to find the corresponding "interdiction set", $A_I \subseteq A_C$, where $k \in A_I$ if and only if k is a forward arc of the minimum cut such that $u_k > \lambda r_k$.

3. Let $R = |A_I|$ and define $x_k(R) = 1$ for all $k \in A_I$ and $x_k(R) = 0$, otherwise. $\mathbf{x}(R)$ solves MXFI(R); see Proposition 1 and Equation (2) above. We have, in effect, optimized the Lagrangian function $z(\lambda,r)$ through LRD(λ,r) for $R = r$. ■

So, we now know that for any fixed value of λ we can obtain a solution to MXFI for some value of R. But we are interested in solving MXFI for all (reasonable) integer values of R. Given how λ affects the solution $\mathbf{x}(R)$ through Proposition 1 and Proposition 2, it is clear that the only values of λ we need be concerned with are in the range $[0,u_{max}]$. Furthermore, from Proposition 2, we can see that the solution $\mathbf{x}(R)$ can only change when the definition of A_I changes; it can be shown that these are the corner points at which the slopes of $z(\lambda,R)$ and $f(\lambda)$ change. Intuitively, this is likely to occur at $\lambda = u_k$ for some k, although examples can be given to show that not all corner points occur at such λ. Nonetheless, we can solve many problems with acceptable accuracy by only evaluating $f(\lambda)$ and $z(\lambda,R)$ at such points. We modify Proposition 2 slightly so that it corresponds to evaluating $f(\lambda)$ only at values $\lambda = u_k$:

Proposition 3: Given a fixed $\lambda = u_k$ for some k, we can solve MXFI(R), for some as yet unspecified R, through LRD(λ,R), as follows:

1. Find a minimum-capacity A_C cut in G with arc capacities $\min\{u_k, \lambda r_k\}\ \forall k \in A$ where $r_k = 1$ for all interdictable arcs k, and $r_k = \infty$, otherwise.

2. Define the interdiction set, $A_I \subseteq A_C$, where $k \in A_I$ if and only if k is a member of A_C such that $u_k \geq \lambda r_k$.

3. Let $R = |A_I|$ and define $x_k(R) = 1$ for all $k \in A_I$ and $x_k(R) = 0$, otherwise. $\mathbf{x}(R)$ solves MXFI(R) and the slope of $z(\lambda,R)$ is zero for all $\lambda \in (u_k - \varepsilon, u_k)$ for some $\varepsilon > 0$. (We might actually need to perturb arc capacities to ensure that such an ε exists.)

4. If $\lambda = u_{k'}$ for some $k' \in A_I$, then let $A_I = A_I - \{k'\}$, $R = R - 1$, and define $x_k(R) = 1$ for all $k \in A_I$ and $x_k(R) = 0$, otherwise. $\mathbf{x}(R)$ solves MXFI(R) and the slope of $z(\lambda,R)$ is zero for all $\lambda \in (u_k, u_k + \varepsilon)$ for some $\varepsilon > 0$ (although arc capacities might need to be perturbed to ensure there exists such an ε). ■

This suggest the following outline of an algorithm for evaluating $\mathbf{x}(R)$ for different values of R:

1. Order the u_k, $u_1 < u_2 < \ldots < u_{|A|}$. These are the potential values for λ (albeit not all-inclusive).

2. Initialize $R = \infty$.

3. For $k = 1, \ldots, |A|$

   a. Let $\lambda = u_k$ and solve the corresponding max-flow problem to evaluate $f(\lambda)$, A_I, R, and $\mathbf{x}(R)$ as in steps 1-3 of Proposition 3.

   b. If $R' \neq R$, print R and $\mathbf{x}(R)$ and let $R' = R$.

   c. If $\lambda = u_{k'}$ for some $k' \in A_I$, then

      i. Let $A_I = A_I - \{k'\}$, $R = R - 1$, and define $\mathbf{x}(R)$ accordingly, all as in step 4 of Proposition 3.

      ii. Print R and $\mathbf{x}(R)$ and let $R' = R$.

   d. If $R = 0$, halt.

The algorithm above first finds a solution $\mathbf{x}(r_1)$ to MXFI(r_1) where r_1 is the number of arcs in a minimum cardinality cut consisting of only interdictable arcs: If the interdictor had r_1 units of interdiction resource, he could reduce the post-interdiction maximum flow to zero. Then, for the same λ we may find a solution to MXFI(r_1−1) as in step 4 of Proposition 3. If not, as λ increases to u_2 or some larger value, a solution $\mathbf{x}(r_2)$ to MXFI(r_2) is found for some $r_2 < r_1$. Unfortunately, it may be that $r_2 < r_1 - 1$ and the algorithm will have failed to solve MXFI(r_1−1). Similarly, the algorithm may solve MXFI(r_h) but fail to solve MXFI(r_{h-1}) and maybe MXFI(r_{h-2}) and so on. This issue will be investigated further, but first we show how multiple arcs with the same capacity are handled.

C. SCALING AND PERTURBING ARC CAPACITIES

We have earlier assumed that all u_k are unique. If they are not unique, we may encounter a network without easily identifiable interdiction sets as Derbes discovered in his work. Derbes' basic method fails to find an optimal solution to MXFI(R) when it cannot identify a set of arcs to interdict that consumes the entire interdiction resource budget exactly. One such failure can occur when optimal cuts found in LRD(λ,R) are composed of a number of arcs with equal capacity. For example, the network in Figure 3 consists of seven interdictable arcs in parallel between the source and the sink with $r_k = 1$ and $u_k = u \ \forall k$. Suppose $R = 4$. There is only one cut, but for $\lambda > 0$, the solution to MXFI(R) from Proposition 1 interdicts all arcs, which is infeasible, or no arcs, rather than the $R = 4$ arcs that are optimal. Thus, Lagrangian relaxation will never find an optimal solution to this problem.

In the context of our algorithm which attempts to solve MXFI(R) for all integer values of R in a certain range, this problem leads to unnecessary gaps in the values of R for which a solution can be obtained. In this example, our algorithm would solve MXFI(R) for $R = 0$ and $R = 7$, but not for any value in between.

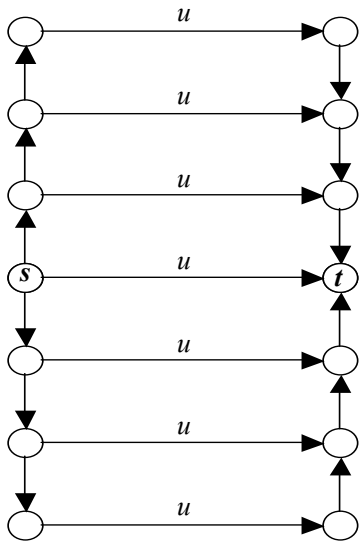

Figure 1. A network without an easily identifiable interdiction set when $r_k = 1$ for all arcs k. All labeled arcs have the same capacity u and are interdictable; all unlabeled arcs have infinite capacity and are uninterdictable.

To cope with this problem, Derbes (1997) scales up all capacities by a factor of 10^5 and then adds a uniform random integer from the range [1,100]. In practice this works well, but it is still possible to have two arcs with identical perturbed capacities, which we would like to avoid.

We deal with the problem of multiple arcs with identical capacities by simply scaling up arc capacities and adding small, unique integer amounts to those arcs with identical capacities. This enables the algorithm is able to differentiate between arcs that would otherwise appear identical. The algorithm we use to accomplish this is

1. Sort the arcs so that $u_1 \leq u_2 \leq \cdots \leq u_{|A|}$;

2. $u_k \leftarrow 10^5 u_k$ for all $k \in A$;

3. For $k = 2$ to $|A|$ { If ($u_k \leq u_{k-1}$) let $u_k \leftarrow u_{k-1} + 1$; }

We store integer arc capacities as "long int" integers in C, which have 32 bits of accuracy and can represent integers somewhat larger than 2.1×10^9. Therefore, our scaled and perturbed capacities will be representable as long as $10^5 u_k + |A| < 2.1\times10^9$. Making

the reasonable assumption that $|A| < 10^8$, we can be assured that capacities as large as $u_k = (2.1\times10^9 - 10^8)/10^5 = 2\times10^4$ are not too large: Our largest capacities are 49. If capacities larger than 2×10^4 are desired, the long int integers can be replaced by 64-bit "long long int" integers. (64-bit arithmetic will be carried out correctly, directly in hardware or indirectly through software, depending on the computer used.)

This scaled-perturbation technique will work fine as long the original arc capacities are not too large—this has just been established for our data—and as long as there are not "too many" arcs with identical capacities. In particular, the technique will not cause an error here unless for two cuts A_{C_1} and A_{C_2}, $\sum_{k\in A_{C_1}} u_k < \sum_{k\in A_{C_2}} u_k$, but $\sum_{k\in A_{C_1}} u'_k > \sum_{k\in A_{C_2}} u'_k$, i.e., the original capacity of cut A_{C_1} is less than the capacity of cut A_{C_2}, but using the scaled and perturbed capacities u'_k reverses the relationship. This cannot happen if the total perturbation added to any minimal cut A_C does not exceed the scale factor of 10^5. An upper bound on the total perturbation added to A_C is $m'|A_C|$ where m' is the maximum number of arcs with identical capacities. Conservative assumptions for our data are that$|A_C| < 60$ and $m' < 600$, so this requirement is satisfied.

After scaling and perturbing arc capacities, the Lagrangian algorithm can be modified to identify, for a given R, certain multiple optimal solutions. In particular, the algorithm might encounter and identify interdiction sets, which have identical original capacities but different perturbed capacities. This would be accomplished in the outlined algorithm by replacing the "if statement" in 3.b with a statement that compares the elements of the previous set A_1 with the current one, and prints out the solution if the two sets are different. For the sake of simplicity, we ignore this possibility.

D. A DETAILED ALGORITHM

In this section, we present a detailed version of the algorithm outlined in section 3.B. This algorithm, Algorithm 1, attempts to solve MXFI(R) for all integers $R \in [0, |A_{\min}|]$, where $A_{\min}$ is a minimum-cardinality cut consisting of only interdictable

arcs. But as stated, it only solves for "normal" values of R, i.e., those values of R that are not problematic.

Algorithm 1 requires the solution of a sequence of maximum-flow problems, and because capacities are non-decreasing in this sequence, the maximum flow $\mathbf{y}'$ in iteration h is feasible for iteration h+1. Thus, it makes sense efficiency-wise to use a flow-augmenting-path max-flow algorithm that maximizes flow starting with the previous iteration's maximum flow; and our implementation of Algorithm 1 does incorporate this feature. However, for the sake of simplicity, we do not show this in our pseudo-code. This maximum-flow algorithm is represented by:

Procedure FindMaxFlow (G, s, t, **u**)

Input: Network $G = (N,A)$, source node s, sink node t
u integer arc capacities $u_k > 0 \ \forall k \in A$

Output: A_C a minimum-capacity cut $A_C \subset A$
y vector of maximum arc flows, $y_k \geq 0 \forall k \in A$
f maximum-flow value

{

This procedure finds a max flow **y**, max-flow value f, and identifies a minimum-capacity cut A_C using a variant of the Edmonds and Karp (1972) max-flow algorithm as implemented by Derbes (1997).

return (A_C , **y**, f)

}

Algorithm 1 can now be stated. Note that solutions are only given in terms of the value for R, the optimal interdiction set A_I given R, and the optimal objective value to MXFI(R) which is $z(R)$.

Algorithm 1: Lagrangian-Relaxation Algorithm for MXFI

Input: Network $G = (N,A)$, source node s, sink node t
u integer arc capacities $u_k > 0 \ \forall k \in A$, all u_k assumed unique
$r_k = 1$ for all interdictable arcs $\forall k \in A$; $r_k = \infty$ otherwise

Output: A_I optimal interdiction set for each normal value of $R \in [0,| A_{\min} |]$;
$z(R)$ optimum objective value to MXFI(R) for each normal value of R;

Step 0: /* define and order the candidate set of λ values */

$\lambda_k \leftarrow u_k \quad \forall k \in A$;

Order the values λ_k such that $\lambda_1 < \lambda_2 < \cdots < \lambda_{|A|}$;

Step 1: Initialize: $R \leftarrow |A|+1, R' \leftarrow |A|+1, f \leftarrow 0, f' \leftarrow 0, \lambda' \leftarrow 0$, $l \leftarrow 1$, $\mathbf{y}' \leftarrow \mathbf{0}$;

Step 2:

While ($l \leq |A|$ and $R \neq 0$){

$\lambda \leftarrow \lambda_l$;

$u'_k \leftarrow \min\{u_k, \lambda r_k\}$ for all $k \in A$; /* adjust arc capacities */

$(A_C, \mathbf{y}, f) \leftarrow$ FindMaxFlow (G, s, t, $\mathbf{u}'$);

$A_I \leftarrow \{k \in A_C \mid y_k = \lambda\}$;

$R \leftarrow |A_I|$;

$z \leftarrow f - \lambda R$;

/* Uncomment the next block to handle problematic R-values */

/*

if ($R - R' > 1$){

Call SolveForProblematic(R, λ, f, R', λ', f', A'_I);

}

* /

print (" The optimal solution to MXFI(R) for R = ", R, "follows: ");

print (The optimal interdiction set and objective value are", A_I, z) ;

if ($u_{k'} = \lambda$ for some $k' \in A_I$){

$A_I \leftarrow A_I - \{k'\}$;

$R \leftarrow R - 1$;

$z \leftarrow z + u_{k'}$;

print (" The optimal solution to MXFI(R) for R = ", R, "follows: ");

print (The optimal interdiction set and objective value are", A_I, z) ;

}

$R' \leftarrow R$, $\lambda' \leftarrow \lambda$, $y' \leftarrow y_a$, $A'_I \leftarrow A_I$, $f' \leftarrow f$, $z' \leftarrow z$;

$l \leftarrow l + 1$;

}

If the change in R-values from one iteration to the next, is never larger than one, Algorithm 1 must solve MXFI(R) for all values of R in the range [0, $|A_{\min}|$] as described in section 3.B. When the change exceeds one, there will be one or more missing solutions and we will modify the algorithm by adding the procedure SolveForProblematic(), to deal with this. This heuristic is described in the next section.

E. PROBLEMATIC VALUES OF *R*

We don't know when problematic R-values will arise, but we know empirically that they are likely to occur and must therefore devise a method to cope with them. We

devise a technique here to evaluate bounds and feasible interdiction sets for the problematic values of R by calling the procedure SolveForProblematic() which is commented out in Algorithm 1. We call the full version of the algorithm Algorithm 2.

When Algorithm 2 finds a solution for $R = r$ and $R = r - 2$, but not for $R = r - 1$, it calls SolveForProblematic() to devise a solution MXFI($r-1$): It "uninterdicts" the arc with the least capacity in the "$R=r$ solution". This yields an upper bound $UB \geq z(r-1)$, of course, and we compute a lower bound $LB = \max\{ f(\lambda_{r-2}) - \lambda_{r-2}(r-1),\ f(\lambda_r) - \lambda_r(r-1) \}$ where λ_{r-2} maximizes $z(\lambda,r-2)$ and λ_r maximizes $z(\lambda,r)$ (and led, respectively, to the solutions of MXFI($r-2$) and MXFI(r)). Absolute and relative optimality gaps are computed for the solution as $AbsGap = UB - LB$ and $RelGap = (100\%) \times AbsGap/LB$.

In practice, Algorithm 2 also creates a heuristic solution for $R = r-1$ by adding an interdiction to the "$R=r-2$ solution." We do not show this in the statement of the algorithm for the sake of simplicity. For the sake of computational efficiency, we have not attempted to optimize the lower bound LB, i.e., we have not tried to maximize $z(\lambda,r-1)$. To do this would require that we explore the region $(\lambda_{r-2}, \lambda_r)$ more fully and solve more maximum-flow problems.

When a sequence of two or more problematic R-values is encountered, we simply apply the above procedure repeatedly; this will be shown by the second example below. In these examples, the notation $A_I(r)$ denotes the interdiction set identified for $R = r$.

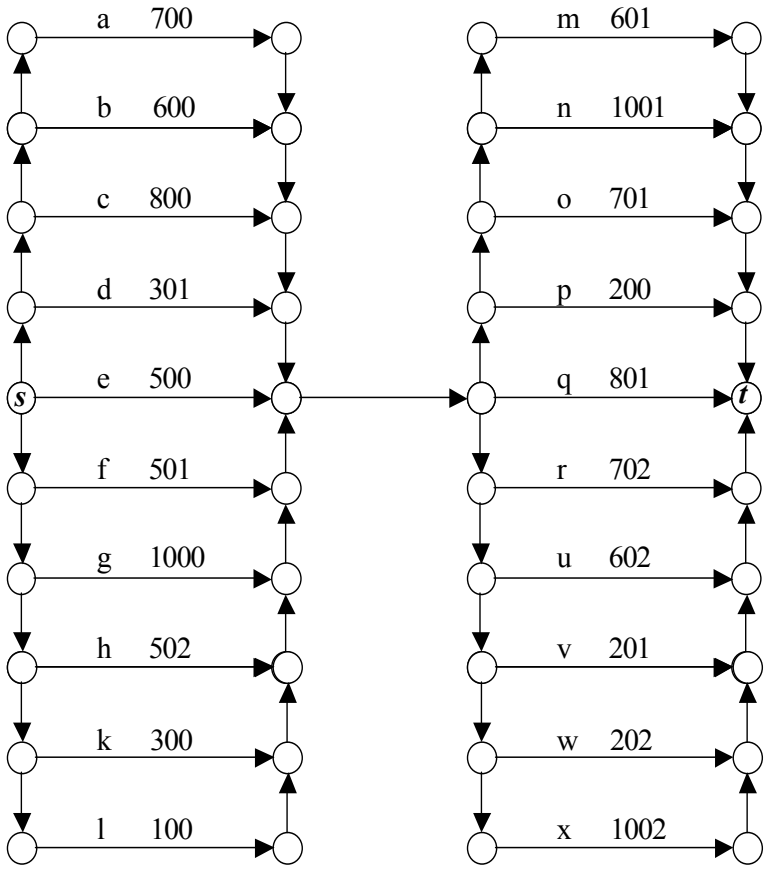

Figure 2. Sample Network, NET5, to Demonstrate Problematic R-values. Numbers on arcs are capacities and letters are labels. Unlabeled arcs cannot be interdicted, i.e., $r_k = \infty$. All labeled arcs have $r_k = 1$.

To illustrate SolveForProblematic(), consider NET5 shown in Figure 2. Algorithm 2 finds a solution for $R = 9$ and then for $R = 7$, but not for $R = 8$. For reference: $z(9) = 100$; $f(\lambda_7) = 2703$, where $\lambda_7 = 300$ optimizes $z(\lambda,7) = z(7)$; the R=9 solution is $A_I(9) = \{a,b,c,d,e,f,g,h,k\}$; and $f(\lambda_9) = 1918$ where $\lambda_9 = 202$. To solve MXFI(8) approximately, we carry out the following steps:

i. $A_I(8) = A_I(9) - k_{\min} = \{a,b,c,d,e,f,g,h,k\} - \{k\} = \{a,b,c,d,e,f,g,h\}$,

ii. $UB = z(9) + u_m = 100 + 300 = 400$,

iii. $LB = \max\{ f(\lambda_7) - \lambda_7 8, f(\lambda_9) - \lambda_9 8\} = \max\{2703-300(8), 1918-202(8)\} = 303$,

iv. $AbsGap = UB - LB = 400 - 303 = 97$, and

v. $RelGap = 100 \times AbsGAP/LB = 32$.

The problem of Figure 2 also provides an example of two consecutive problematic R-values. In particular, Algorithm 2 solves exactly for $R = 7$ ($\lambda_7 = 502$) and for $R = 4$ ($\lambda_4 = 600$), but not for $R = 6$ and $R = 5$. For reference, $z(7) = 603$; $f(\lambda_7) = 4117$, $f(\lambda_4) = 4604$, where $\lambda_4 = 600$ optimizes $z(\lambda,4) = z(4)$; and $A_1(7) = \{m,n,o,q,r,u,x\}$. Then, the algorithm will compute:

1. For $R = 6$,

   i. $A_1(6) = \{m,n,o,q,r,u,x\} - k_{min} = \{m,n,o,q,r,u,x\} - \{m\} = \{n,o,q,r,u,x\}$,

   ii. $UB = z(7) + u_m = 603 + 601 = 1204$,

   iii. $LB = \max\{f(\lambda_4) - \lambda_4 6, f(\lambda_7) - \lambda_7 6\} = \{4604 - 600(6), 4117 - 502(6)\} = 1105$,

   iv. $AbsGap = UB - LB = 1204 - 1105 = 99$, and

   v. $RelGap = 100 \times AbsGAP/LB = 8.9$.

2. For $R = 5$,

   i. $A_1 = \{n,o,q,r,u,x\} - k_{min} = \{n,o,q,r,u,x\} - \{u\} = \{n,o,q,r,x\}$,

   ii. $UB = z(6) + u_u = 1204 + 602 = 1806$,

   iii. $LB = \max\{f(\lambda_4) - \lambda_4 5, f(\lambda_7) - \lambda_7 5\}$ $\{4604 - 600(5), 4117 - 502(5)\} = 1607$,

   iv. $AbsGap = UB - LB = 1806 - 1607 = 199$, and

   v. $RelGap = 100 \times AbsGAP/LB = 12.4$.

This example also illustrates that our methodology does not necessarily compute the best lower bound. For instance, the lower bound we obtain for $R = 5$ is 1607, but the best lower bound is $\max_\lambda z(\lambda,5) = 1670$ which occurs at $\lambda = 535$. Algorithm 2 will never discover this because it only evaluates $z(\lambda,R)$ for λ equal to some arc capacity; and 535 is not an arc capacity in this example.

SolveForProblematic( ) is now specified in detail here:

Procedure SolveForProblematic (R, λ, f, R', λ', f', A_I')

Input:

R	current R-value correctly solved
λ	value of parameter that maximized $z(\lambda,R)$ to solve MXFI(R)
f	maximum flow $f(\lambda)$
R'	previous R-value correctly solved for, ($R' > R + 1$)
λ'	value of parameter that maximized $z(\lambda,R')$ to solve MXFI(R')

Output:

A_I'	interdiction sets that are approximate solutions for problematic R-values in the range $R'-1, R'-2, \ldots, R+1$
UB	upper bound for problematic value of R
LB	lower bound value for problematic value R
$AbsGAP$	absolute optimality gap $UB - LB$
$RelGAP$	relative optimality gap $((100\%)\times(UB - LB)/LB$

Step 0:

$UB \leftarrow z'$;

Step 1:

For ($r = R' - 1$ down to $R + 1$){

$k_{\min} \leftarrow \underset{k \in A_I'}{\operatorname{argmin}} u_k$;

$LB \leftarrow \max\{ f - \lambda r, f' - \lambda' r \}$;

$UB \leftarrow UB + u_{k_{\min}}$;

$AbsGAP \leftarrow UB - LB$;

$RelGAP \leftarrow 100\times AbsGAP/LB$;

$A_I' \leftarrow A_I' - k_{\min}$;

print (" For problematic value R = ", r, "the approximate soln. is: ");

print (A_I' ,UB, LB, $AbsGAP$, $RelGAP$);

}

E. COMPLEXITY OF THE ALGORITHM

Finding the minimum-capacity cut after finding the maximum-flow requires $O(|A|)$ time at worst. The sort of the values for λ_k has complexity $O(m\log m)$. If no problematic values of R are observed, the work in the algorithm is clearly dominated by the need to solve perhaps as many as m maximum-flow problems. In this case, the complexity of the overall algorithm is $O(m\mathrm{g}(m,n))$ where the max-flow algorithm has complexity $O(\mathrm{g}(m,n))$. At most m problematic values of R can be encountered and the work required to deal with these is at most $O(m)$ each. Thus, the overall complexity of dealing with problematic R-values is $O(m^2)$. The overall complexity remains $O(m\mathrm{g}(m,n))$ since $O(\mathrm{g}(m,n))$ is certainly worse than $O(m)$. In practice, we may achieve better

performance than this because we are solving a sequence of related maximum flow problems, and not solving each problem from scratch.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. COMPUTATIONAL RESULTS

We describe the networks tested in the first part of this chapter and present computational results in the second part.

A. TEST NETWORK DESIGN

We have tested our algorithm using five artificial, directed networks; see Table 1. The structure of these networks should be similar to the structure of road networks of interest. These networks are based on $n_1 \times n_2$ grid networks where n_1 is the number of nodes on the horizontal axis and n_2 is the number of nodes on the vertical axis. Horizontal arcs are oriented from "west to east" and all nodes directly above or below another are connected with a pair of anti-parallel arcs. Anti-parallel arcs are separately interdictable. We also have arcs oriented in the southeast and northeast directions. All westernmost nodes are connected to a source node with artificial, uninterdictable, infinite-capacity arcs and all easternmost nodes are connected to a sink node with artificial, uninterdictable, infinite-capacity arcs. The arc capacities u_k are randomly drawn from the discrete uniform distribution on [1,49] and $r_k = 1$ is assumed for all interdictable arcs k.

We base our implementation of Algorithm 2 on Derbes (1997), which uses an improved version of the Edmonds and Karp (1972) maximum-flow algorithm. His code only attempts to solve MXFI(R) for single values for R; ours attempts to solve for all values $R \in [0, |A_{\min}|]$ where $A_{\min}$ is a minimum-cardinality cut in the network consisting of only interdictable arcs.

Network	n_1	n_2	n	m	f^*
NET27	5	5	27	86	238
NET51	7	7	51	188	361
NET101	11	9	101	412	414
NET122	8	15	122	499	765
NET402	20	20	402	1826	1080

Table 1. Statistics for Test Networks. n is the total number of nodes and m the total number of nodes. The networks are based on $n_1 \times n_2$ grids of nodes. The number of interdictable arcs is $m - 2n$. f^* is the uninterdicted maximum flow, where arc capacities are uniformly distributed random integers in the range [1,49].

B. RESULTS

Test results are summarized in Table 2, which explicitly shows only the results for problematic R-values since MXFI(R) is solved exactly for other, normal values. We are primarily interested the percentage of problematic R-values and the accuracy of results for such values. The maximum flow value without interdiction is indicated by f^* in the tables. The maximized value of the Lagrangian function is $z(\lambda,R)$ while $\overline{z}(\lambda,R)$ is the upper bound value, which equals $z(\lambda,R)$ when R is not problematic but is obtained from the heuristic when it is.

Table 2 shows that no problematic R-values occur for the first two problems and that, in general, problematic values occur less than 25% of the time. Furthermore, absolute optimality gaps are always small, being at most 3% of the uninterdicted maximum flow. NET402 exhibits 10 out of 58 R-values as problematic all of which have relative optimality gaps of less than 2.7%. Large relative optimality gaps, 33.3% and 20.0%, occur in two instances for NET122, but these also depend on the small value of

the lower bound. The rest of the problematic *R*-values for this network have relative gaps of less than 1.1%, which seem very reasonable.

In Tables 3 and 4, we explore changes in optimality gaps for one of the test networks, NET122, as a function of the coarseness of the arc-capacity distribution. When the capacity distribution is coarse, i.e., there are few arc capacities, the curves for $z(\lambda,R)$ will not be evaluated at many points and we can expect that lower bounds will not be very accurate. Furthermore, fewer potential solutions will be seen and thus we expect upper bounds to be worse. We note that the largest problem, NET402, requires 384 seconds to solve if each maximum-flow problem is solved from scratch, so solving the maximum-flow problem in iteration h+1 starting with the flow from iteration h, as we do, can yield a substantial improvement in run time.

In Table 3, NET122 is evaluated with capacities randomly drawn from {16,32}, {8,16,24,32}, and {4,8,12,16,20,24,28,32} (denoted 16[1,2], 8[1,4], 4[1,8], respectively); in Table 4 it is evaluated with capacities drawn from {2,4,6…,32}, and {1,2,…,32} (denoted 2[1,16], [1,32], respectively). As suspected, these tables indicate that more coarsely distributed arc capacities lead to poorer results: The coarsest distribution has the largest absolute and relative errors and the least coarse the smallest. It also appears from the results that when there is a non-zero optimality gap, a lower bound on this gap is the smallest non-zero difference between arc capacities. In particular, the smallest non-zero gap for 16[1,2] is 16, for 8[1,4] is 8, and so on. Also, we see that gaps appear to be monotonically non-decreasing in a contiguous sequence of problematic *R*-values. For instance, consider the sequence of R-values from 19 down to 10 in 16[1,2] and similar sequences in the other problems.

Network	f^*	R	Run Time (cpu sec.)	Problematic R -Values	$\bar{z}(\lambda,R)$	$z(\lambda,R)$	Abs. Gap	Rel. Gap
NET27	238	[1,13]	0.1	none	NA	NA	NA	NA
NET51	361	[1,19]	0.5	none	NA	NA	NA	NA
NET101	414	[1,25]	1.0	13	75	73	2	2.7
NET122	765	[1,43]	1.0	41	4	3	1	33.3
				40	6	5	1	20.0
				23	166	165	1	0.6
				22	184	182	2	1.1
				21	200	199	1	0.5
				17	277	276	1	0.4
				16	299	297	2	0.7
				15	320	319	1	0.3
				11	410	408	2	0.5
				10	432	431	1	0.2
NET402	1080	[1,58]	16.0	47	37	36	1	2.7
				46	42	41	1	2.4
				38	113	112	1	0.9
				33	175	173	2	1.1
				32	190	186	4	2.1
				31	202	200	3	1.0
				22	350	349	1	0.3
				21	373	371	2	0.5
				20	396	393	3	0.8
				19	417	416	1	0.2

Table 2. Summarized Test Results for the Five Test Networks. f^* is the uninterdicted value for maximum flow. "NA" indicates "not applicable," used because MXFI(R) solves exactly for all values of R for NET27.

u_k	f^*	R	Run Time (cpu sec.)	Problematic R -Values	$\bar{z}(\lambda,R)$	$z(\lambda,R)$	Abs. Gap	Rel. Gap
16[1,2]	912	[1,43]	1.5	19	400	384	16	4.2
				18	432	400	32	8.0
				17	464	416	48	11.5
				16	496	432	64	14.8
				15	528	448	80	17.8
				14	544	464	80	17.2
				13	560	496	64	12.9
				12	576	528	48	9.9
				11	592	560	32	5.7
				10	608	592	16	2.7
8[1,4]	672	[1,43]	1.5	28	128	120	8	6.7
				27	136	128	8	6.3
				18	280	272	8	2.9
				17	304	288	16	5.5
				16	328	304	24	7.9
				15	344	320	24	7.5
				14	360	336	24	7.1
				13	376	352	24	6.8
				12	392	368	24	6.5
				11	408	384	24	6.3
				10	424	408	16	3.9
				9	440	432	8	1.8
4[1,8]	584	[1,43]	1.5	21	172	168	4	2.4
				14	284	280	4	1.4
				13	300	296	4	1.4
				12	316	312	4	1.3
				11	332	328	4	1.2
				10	348	344	4	1.2
				9	364	360	4	1.1
				3	496	492	4	0.8

Table 3. Partial Test Results for NET122 with Restricted Arc Capacities. 16[1,2] indicates capacities uniformly distributed from the set {16,32}, 8[1,4] from the set {8,16,24,32} and 4[1,8] from the set {4,8,…,32}.

u_k	f^*	R	Run Time (cpu sec.)	Problematic R-Values	$\bar{z}(\lambda,R)$	$z(\lambda,R)$	Abs. Gap	Rel. Gap
2[1,16]	530	[1,43]	1.5	21	144	142	2	1.4
				20	156	154	2	1.3
				16	212	210	2	1.0
				15	226	224	2	0.9
				14	240	238	2	0.8
				11	288	286	2	0.7
[1,32]	503	[1,43]	1.5	22	121	120	1	0.8
				21	133	132	1	0.8
				16	199	197	1	1.0
				15	213	211	2	0.9
				14	226	225	1	0.4
				13	240	239	1	0.4
				12	254	253	1	0.4
				11	268	267	1	0.4

Table 4. Partial Test Results for NET122 with Restricted Arc Capacities. 2[1,16] indicates capacities uniformly distributed from the set {2,4,…,32} and [1,32] from the set {1,2,…,32}.

V. CONCLUSIONS AND FUTURE WORK

A. SUMMARY

This thesis has studied a special solution technique for solving or approximately solving the maximum-flow network-interdiction problem (MXFI). In MXFI, a network user attempts to maximize flow of a commodity through the network, while an interdictor destroys network arcs, using limited interdiction resources, to minimize this maximum flow.

MXFI can be modeled as a binary integer problem and solved but this approach can be slow. A Lagrangian-relaxation technique developed by Derbes (1997) has shown promise for solving the problem more quickly, and this thesis has extended Derbes' technique for solving MXFI for all values of R in a specified range. It is assumed that one unit of resource is required to interdict any interdictable arc, so we are concerned only with integer values of R.

In effect, our basic algorithm, Algorithm 1, determines the breakpoints in the piecewise-linear, concave Lagrangian function, which can only occur when the Lagrangian multiplier λ equals the capacity of some arc. For each linear piece of the function, MXFI is solved exactly for some value of R, with increasing values of λ corresponding to smaller values of R. But the algorithm may leave gaps in the sequence of R-values for which the problem is solved. Algorithm 2 combines Algorithm 1 and a heuristic to fill in these gaps, usually quite successfully.

We have performed basic testing of Algorithm 2, coded in C, using five test networks ranging in size from 27 nodes and 86 arcs to 402 nodes and 1826 arcs. Arc capacities are uniformly distributed random integers in the range [1,49]. All tests are performed on a 700 MHz Pentium III personal computer with The Microsoft Windows Millennium operating system and Microsoft Visual C++ compiler. The largest network is solved in 16 seconds. Absolute optimality gaps are always small, at most two percent of the uninterdicted maximum flow. Relative gaps are typically small, a few percent, but can be large if the post-interdiction maximum flow is small.

Additional testing evaluated one of the networks with capacities distributed more and less coarsely. In particular, capacities are drawn randomly from {16,32}, {8,16,24,32},…,{1,…,32}. We conclude that absolute and relative optimality gaps tend to be larger with the coarser capacity distributions.

B. FUTURE WORK

Substantial efficiency is gained in our solution procedure by solving the maximum-flow problem in iteration h+1 starting with the solution from iteration h; flow-augmenting-path maximum-flow algorithms work well in this context. However, there exist more efficient flow-augmenting-path algorithms than ours, and it might be worth investigating such algorithms for the purpose of improving overall run times.

Further research on this problem may focus on the solution methods of problematic R-values. We have devised a heuristic to deal with these, but an exact branch-and-bound algorithm could exploit the lower bounds provided by our Lagrangian algorithm. We also know that our lower bounds are not always maximal, and this issue needs to be investigated further: Improved bounds could be obtained by simply defining a finer grid for λ than the one we use (which corresponds to arc capacities), but of course this would add to the computational burden. The finer grid might also lead to better feasible solutions, too. It may also be possible to reformulate the problem so that Lagrangian relaxation yields smaller optimality gaps. Of course, the technique should be generalized so that the resource required to interdict an arc, r_k, is a general integer rather than restricting $r_k = 1$.

The model MXFI can be extended to dynamic networks by adding traversal time for each arc in the networks. The model can also be extended to consider stochastic arc capacities and/or uncertain interdiction successes (Cormican, Morton and Wood 1995). Perhaps our Lagrangian technique could be used in the solution procedure for these problems.

Golden (1978), Israeli (1999), Israeli and Wood (1999) and Wevley (1999) study another category of network interdiction model where an interdictor attempts to interdict (destroy or lengthen) arcs, using limited interdiction assets, to maximize the length of a

shortest *s-t* path. While these models are similar to MXFI, that earlier work is not relevant to this thesis. However, the Lagrangian-relaxation technique developed here may have an analog for shortest-path interdiction and this may be worth investigating.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Ahuja, R.K., Magnanti, T.L., and Orlin, J. B., *Network Flows*, Prentice Hall, New Jersey, 1993.

Akgun, I., "The K-Group Maximum-Flow Network Interdiction Problem," Master's Thesis, Naval Postgraduate School, Monterey, California, March 2000.

Cormican, K.J., "Computational Methods for Deterministic and Stochastic Network Interdiction Problems," Master's Thesis, Naval Postgraduate School, Monterey, California, March 1995.

Cormican, K.J., Morton, D.P., and Wood, R.K., "Stochastic Network Interdiction," *Operations Research*, Vol. 46, pp. 184-197, 1998.

Derbes, H.D., "Efficiently Interdicting A Time-expanded Transshipment Network," Master's Thesis, Naval Postgraduate School, Monterey, California, September 1997.

Durbin, E.P., "An Interdiction Model of Highway Transportation," Report RM-4945-PR, RAND Corporation, Santa Monica, California, May 1966.

Edmonds, J., and Karp, R.M., "Theoretical Improvements in Algorithmic Efficiency for Network flow Problems," *Journal of the Association for Computing Machinery*, Vol. 19, No. 2, April 1972.

Ford, L.R., Jr., and Fulkerson, D.R., "Maximal Flow through A Network," *Canadian Journal of Mathematics*, 1956.

Ghare, P.M., Montgomery, D.C., and Turner, W.C., "Optimal Interdiction Policy for a Flow Network," *Naval Research Logistics Quarterly*, Vol. 18, pp. 37-45, 1971.

Golden, B., "A Problem in Network Interdiction," *Naval Research Logistics Quarterly*, Vol. 25, pp. 711-713, 1978.

Helmbold, R.L., "A Counter Capacity Network Interdiction Model," Report R-611-pr, RAND Corporation, Santa Monica, California, March 1971.

Israeli, E., "System Interdiction and Defense," Doctoral Dissertation, Naval Postgraduate School, Monterey, California, March 1999.

Israeli, E., and Wood, R.K., "Shortest Path Network Interdiction," (in review).

Lubore, S.H., Ratliff, H.D., and Sicilia, G.T., "Finding the n most Vital Links in Flow Networks," *Management Science*, Vol. 21, No. 5, 1975.

McMasters, A.W., and Mustin, T.M., "Optimal Interdiction of a Supply Network," *Naval Research Logistics Quarterly*, Vol. 17, pp. 261-268, 1970.

Reed, B.K., "Models for Proliferation Interdiction Response Analysis," Master's Thesis, Naval Postgraduate School, Monterey, California, September 1994.

Phillips, C.A., "The Network Destruction Problem," Report SAND-92-0186C, Sandia National Laboratories, Albuquerque, New Mexico, April 27, 1992.

Steinrauf, R.L., “Network Interdiction Models,” Master’s Thesis, Naval Postgraduate School, Monterey, California, September 1991.

Wevley, C.M., “The Quickest Path Network Interdiction Problem,” Master’s Thesis, Naval Postgraduate School, Monterey, California, March 1999.

Whiteman, P.S., “Improving single strike effectiveness for network interdiction,”

Military Operations Research, Vol. 4, pp.15-30, 1999.

Wollmer, R.D., “Removing Arcs from a Network,” *Operations Research*, Vol. 12, pp. 807-1076, 1964.

Wollmer, R.D., “Algorithm for Targeting Strikes in a Lines-of-Communication Network,” *Operations Research*, Vol. 18, pp.497-515, 1970.

Wood, R.K., ”Deterministic Network Interdiction,” *Mathematical and Computer Modeling*, Vol. 17, No. 2, pp. 1-18, 1993.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox LibraryNaval Postgraduate School
   Monterey, CA

3. Professor R. Kevin Wood
   Department of Operations Research, Code OR/Wd
   Naval Postgraduate School
   Monterey, CA

4. LCDR Kelly J. Cormican.
   Department of Operations Research, Code OR/Wd
   Naval Postgraduate School
   Monterey, CA

5. Deniz Kuvvetleri Komutanligi
   Kutuphane
   Bakanliklar, Ankara, TURKEY

6. Deniz Harp Okulu Komutanligi
   Kutuphane
   Tuzla, Istanbul, TURKEY

7. Bilkent Universitesi Kutuphanesi
   Bilkent, Ankara, TURKEY

8. Orta Dogu Teknik Universitesi Kutuphanesi
   Balgat, Ankara, TURKEY

9. Bogazici Universitesi Kutuphanesi
   Bebek, Istanbul, TURKEY

10. Dz.Utgm. Levent Bingol
    Deniz Kuvvetleri Komutanligi
    Bakanliklar, Ankara, TURKEY